

**Quaternion simulation of two level entangled state product calculation verification adapted from  
“Quantum Correlations are Weaved by the Spinors of the Euclidean Primitives” III B 1.  
By John Reed; modified by Fred Diether to work with 3D vectors and no hidden variable.**

```
In[553]:= << Quaternions`;
β0 = Quaternion[1, 0, 0, 0];
β1 = Quaternion[0, 1, 0, 0];
β2 = Quaternion[0, 0, 1, 0];
β3 = Quaternion[0, 0, 0, 1];
Qcoordinates = {β1, β2, β3};
s = 0; t = 0; u = 0;
m = 50000;
plotArray = Table[{0, 0}, m];
For[nn = 1, nn ≤ m, nn += 1,
{vectorA = RandomPoint[Sphere[]];
vectorB = RandomPoint[Sphere[]];
vectorS = RandomPoint[Sphere[]]; (*Singlet spin vector*)
Da = vectorA.Qcoordinates;
Db = vectorB.Qcoordinates; (*Convert to quaternion coordinates*)
Ls = vectorS.Qcoordinates; (*singlet spin quaternion*)
If[vectorA.vectorS > 0, Lsa = Da, Lsa = -Da];
If[vectorB.vectorS > 0, Lsb = Db, Lsb = -Db];
A = Re[FromQuaternion[Da ** Lsa]]; (*A detector*)
B = Re[FromQuaternion[Db ** Lsb]]; (*B detector*)
Aq = -Da ** Ls;
Bq = Ls ** Db;
q = 0;
q = (Aq ** Bq); (*Product calculation*)
s = s + q;
t = t + A;
u = u + B;
φA = ArcTan[vectorA[[2]], vectorA[[1]]];
φB = ArcTan[vectorB[[1]], vectorB[[2]]];
If[φA * φB > 0, angle = ArcCos[vectorA.vectorB] / Degree,
angle = (-ArcCos[vectorA.vectorB]) / Degree];
plotArray[[nn]] = {angle, Re[q]}];
mean = FromQuaternion[s / m] (*shows vanishing of the non-real part K*)
avA = t / m;
avB = u / m;
Print["<A> = ", avA, " <B> = ", avB]
simulation = ListPlot[plotArray, PlotMarkers → {Automatic, Small}];
negcos = Plot[-Cos[x Degree], {x, -180, 180}, PlotStyle → {Red}];
Show[simulation, negcos ]
Out[563]= (-0.00467045 + 0.00120702 I) - 0.00168178 J - 0.00121973 K
<A> = 0.00468 <B> = 0.00248
```

